Data Warehouse & ETL Offload Code Samples Overview and Install Guide

The Data Warehouse & ETL Offload Code Samples provide sample code artifacts to support data warehousing and ETL offload solution patterns in the Oracle Public Cloud and in an Oracle Cloud at Customer deployment. The code artifacts and sample dataset provided are described below by folder. Installation and execution guidelines are included below.

Code Samples & Installation Content

Pre-Requisites and Assumptions 1
Global Files - SampleSourceFiles 2
Loading Autonomous Data Warehouse 3
GoldenGate_Parameter_Samples 3
ODISampleAutonomousDataWarehouseCloudLoad 3
BigDataCloudETLOffloadSparkNotebook 4
CloudAtCustomerODIETLOffload 5

Pre-Req's and Assumptions

These code samples are built using Oracle Cloud services (either Oracle Public Cloud or Oracle Cloud at Customer) and are provided as-is with no expressed warranty. Complete documentation on the Data Warehouse and ETL Offload Solution patterns are available online at <LINK HERE> and should be read prior to implementing these samples. The solution documentation provides step-by-step guides to configuring the services and should be followed prior to installation / execution of the sample workloads. To implement these samples you will need to have provisioned the following services –

Oracle Autonomous Data Warehouse Cloud

Oracle Data Integration Platform Cloud

Oracle Analytics Cloud

Oracle Big Data Cloud (for Oracle Public Cloud ETL Offload)

Oracle Big Data Cloud @ Customer (for Cloud@Customer ETL Offload)

Global Files - SampleSourceFiles

SampleSourceFiles provide a set of .csv files that are used throughout the solution. These files are used in both ODI Smart export samples (loading ADWC and ETL Offload for BigDataCloud@Customer) as well as in the sample BigDataCloud Notebook for ETL Offload / Spark processing in the Oracle Public Cloud

CUSTOMER_SRC_FILE.csv provides a set of made up customer data (customer names are random letters and numbers). This is used to load the CUSTOMER dimension and to provide REGION data for the ETL Offload spark sample

Column Ordinal	Column Name	DataType
1	CUSTOMER_ID	NUMERIC
2	FIRST_NAME	VARCHAR
3	LAST_NAME	VARCHAR
4	REGION	VARCHAR
5	CITY	VARCHAR

SRC_PRODUCT.csv provides a set of product data. This is used to load the PRODUCT dimension in ADWC and to provide Product Category / Family data for the ETL Offload Spark sample.

Column Ordinal	Column Name	DataType
1	PRODUCT_ID	NUMERIC
2	PRODUCT_NAME	VARCHAR
3	PRICE	NUMERIC
4	FAMILY	VARCHAR

ORDERS_FILE.csv provides a set of order data. This is used to load the SALES_FACT fact table in ADWC (and the subsequent SALES_ANALYSIS table). It provides the dataset for the SALES_ANALYSIS ETL Offload Spark Notebook and ODI project.

Column Ordinal	Column Name	DataType
1	ORDER_ID	NUMERIC
2	CUSTOMER_ID	NUMERIC
3	ORDER_DATE	DATE
4	PRODUCT_ID	NUMERIC
5	AMOUNT	NUMERIC
6	QTY	NUMERIC
7	ORDER_LINE_NUMBER	NUMERIC

Loading Autonomous Data Warehouse

This folder provides artifacts demonstrating how to replicate a source table to ADWC for reporting with GoldenGate and how to load a star schema data warehouse with ODI. (Note this project can also be used against an Exadata Cloud at Customer solution but may require you to change the knowledge modules used in the solution)

GoldenGate_Parameter_Samples

Artifacts in this folder represent sample GoldenGate parameter files for real-time replication of data from an Oracle database source (on-premise or DBCS) to an ADWC target. These files can be used with any GoldenGate for Oracle 12.3 implementation including the Data Integration Platform Cloud's REMOTE AGENT.

exadwc.prm is a sample GoldenGate EXTRACT parameter file to capture from an Oracle database source. To configure this file replace the <SID> entry with your Oracle database SID or CDB name. This file is configured to capture from the schema ADWC_SRC.

padwc.prm is a sample GoldenGate EXTRACT PUMP parameter file to pump transactions capture via the extract to the Data Integration Platform Cloud instance for delivery to Autonomous Data Warehouse Cloud. To configure this file – replace the <remote_host_ipt> and <SID> entries.

radwc.prm is a sample GoldenGate REPLICAT parameter file for a Classic replicat. This applies the transactions to the Autonomous Data Warehouse Cloud instance. To configure this file replace the <SID> entry with your SID or CDB name. This applies data from the ADWC_SRC schema to the adwc_repl schema in your adwc instance.

Place these files in your dirprm directory for the GoldenGate installation. (exadwc.prm and padwc.prm in the source and radwc.prm in your DIPC host / DIPC remote agent installation).

Follow the instructions in the Data Warehouse Solution Pattern for info on configuring GoldenGate within the DIPC remote agent and where to leverage these files.

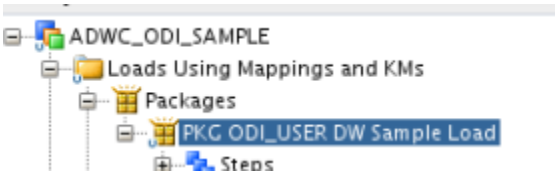
ODISampleAutonomousDataWarehouseCloudLoad

These artifacts provide create user / ddl scripts to create a small star schema in the Autonomous Data Warehouse Cloud and an ODI Smart Export to load the Autonomous Data Warehouse Cloud star schema.

To import / execute these loads you must perform the following steps –

- Connect to your ADWC instance via SQL Developer
- Run the CreateODI_USER_and_TABLES.sql script to create the ODI_USER and the target tables for the ODI project. ***Note – replace the <PASSWORD> tag in the first line of the script with the password you wish to use for the ODI User.*
- SSH into your Data Integration Platform Cloud instance as OPC user

- Create a directory in your DIPC instance
    - Mk dir /tmp/ADWCSample
- Copy the SourceFiles ORDERS_FILE.csv, SRC_PRODUCT.csv and CUSTOMER_SRC_FILE.csv and ODIADWCSample.xml to your DIPC instance – the /tmp/ADWCSample directory you just created
- Change permissions on the directory and files to ensure the oracle user can read the files
    - Sudo chmod 755 -R /tmp/ADWCSample
- VNC into your DIPC instance.
- Run ODI Studio
- Use Smart Import to import the ODI processes.
- Navigate to the Topology Manager in ODI Studio and alter the ADWC connection strings (review the Solution Documentation for more details on how to obtain / configure your ODI / ADWC connection).
- Open the ODI project and review the mappings.
- To run the processes execute the ODI package PKG ODI_User DW Sample Load



- Using SQL Developer – query the tables in the ADWC ODI_USER schema
    - CUST_DIM
    - DATE_DIM
    - PRODUCT_DIM
    - SALES_FACT
    - SALES_FACT_ANALYSIS

Review the ODI mappings. Note that the 2 folders provide 2 methods for executing dimension and fact loads in ADWC:

Loads Using Mappings and KM's provides mappings the leverage certified Knowledge Modules for ADWC. These are appropriate for Type 1 dimension style loads.

Loads using Dimensions and Cubes objects provides mappings that leverage the Dimensions and Cubes objects within ODI to load the same set of tables into the ADWC ODI_USER schema. The Dimensions objects are appropriate for Type 2 or Type 3 dimension loads.

BigDataCloudETLOffloadSparkNotebook

This artifact is a sample BigDataCloud Notebook that demonstrates Spark to load data from files stored in Oracle Object Storage – perform an ETL routine leveraging SparkSQL and then store the result in multiple file formats back in Object Storage (all running in the Oracle Public Cloud).

To configure and execute the Spark notebook –

- Login to your Oracle Public Cloud acct.
- Create a Container within your Oracle ObjectStorage service (for example BDCETL)

- Navigate to your BigDataCloud instance and the Notebook tab.
- Click Import to import the notebook (point to the json file and import).
- Open the notebook and click the run / play button to execute the Spark script.
- The Spark notebook generates a SALES_FACT_ANALYSIS similar to the results of the ODI ADWC processes.

To review the results simply execute the %SQL section of the notebook which queries the resulting SALES_ANALYSIS table in BigDataCloud.

CloudAtCustomerODIETLOffload

These artifacts demonstrate ODI performing ETL Offload in a BigData Cloud@Customer (Cloudera) environment. These leverage the SourceFiles sample dataset to ingest data into the Cloudera Cluster (via SQOOP or Spark) and then ODI executes Spark workloads to generate the SALES_ANALYSIS data set.

To configure and execute the ODI jobs –

- Follow the documentation on configuring your BigData Cloud at Customer machine for ODI workload. Do not create your topology objects yet however.
- Ssh / vnc into you ODI Agent node in your cluster
- Create a directory in your clusters local file system
    - Mkdir /tmp/sourcefiles
- Copy the ODISmartExport_ETLOffload_BigDataCloud@Customer.zip file to your ODI agent node in your cluster.
- Copy the SourceFiles ORDERS_FILE.csv, SRC_PRODUCT.csv and CUSTOMER_SRC_FILE.csv to your DIPC instance – the /tmp/sourcefiles directory you just created
- Run ODI Studio
- Smart Import the ODI project into you environment
- Configure your topology connections based on the Solution documentation step by step guide.
- Edit the ORCL_SRC connection to point to an Oracle database that you have (the ingestion mappings pull from an Oracle db to demonstrate how to leverage either SQOOP or Spark to ingest data to your cluster)
- Follow the Solution documentation instructions on configuring your Hadoop credential store for the ORCL_SRC connection if you plan on using Spark to ingest the data
- Review the ODI project.
- 0. Create Oracle Source Objects
    - The PKG Create Oracle Objects and Load Data will create tables in your source Oracle database and load the SourceFiles to that database.
- 1. Ingest Data to BigData HDFS – SQOOP & Spark
    - The PKG Ingest Data to Hive leverages ODI mappings to import data into your Big data cluster leverage SQOOP.
    - Open one of the mappings and navigate to the physical tab. Note that there are physical designs for both SQOOP and Spark ingestion. Either can be leveraged to move data into the cluster.

- 2. ETL Offload – Spark
    - The Mapping MAP Spark ETL Offload… demonstrates how to create a mapping in ODI that leverages Spark to perform ETL offload to the cluster and create a SALES_ANALYSIS data set.
- Execute the packages / mappings listed above in order to
    - Create Oracle source objects
    - Ingest data into your cluster
    - Generate a SALES_ANALYSIS dataset leveraging ODI's Spark capabilities.